1. An apparatus comprising:

at least one processor;

a memory coupled to the at least one processor; and

a scheduling manager residing in the memory;

the scheduling manager dynamically managing access of work items of a program to additional computer resources other than the at least one processor based on determining if valuations of each work item to be processed exceed estimated processor costs for respective ones of the work items.

2. The apparatus of claim 1 wherein the scheduling manager applies a valuation heuristic to each work item.

3. The apparatus of claim 1 wherein the processing of each work item is delayed if an estimated processing cost of each work item exceeds the valuation of the respective work item.

4. The apparatus of claim 3 wherein the scheduling manager applies a priority algorithm for preventing starvation of computer resources to those work items which have been delayed, whereby the processing of all the work items in a program is completed.

5. The apparatus of claim 4 wherein the priority algorithm increases respective valuations of delayed work items so as to complete processing of each of the work items prior to or at a cut-off processing date of the work item.

6. A computer-implemented method for managing access to computer resources based on valuations of work items of a program relative to their respective estimated processing costs, the method comprising: providing a scheduling manager; and, dynamically managing the access of additional computer resources by respective ones of the work items if the valuation of each

of the work items exceeds the cost of processing corresponding ones of the work items.

7. The method of claim 6 further comprising applying a valuation heuristic to each work item.

8. The method of claim 6 further comprising applying a priority algorithm for preventing starvation of computer resources to those work items which have been delayed, whereby the processing of all the work items in a program is completed.

9. The method of claim 7 further comprising having the priority algorithm increase respective valuations of delayed work items so as to complete processing of each of the work items prior to or at a cut-off processing date of the work item.

10. A program product comprising:

a scheduling manager that dynamically manages the access of additional computer resources by respective ones of work items of a program if the valuation of each of the work items exceeds the cost of processing corresponding ones of the work items; and,

a computer readable signal bearing media bearing the scheduling manager.

11. The program product of claim 10 wherein the scheduling manager applies a valuation heuristic to each work item to establish a valuation for each of the work items.

12. The program product of claim 10 wherein the scheduling manager dynamically delays each work item if an estimated cost of processing exceeds the valuation of the respective work item.

13. The program product of claim 10 wherein the scheduling manager applies a priority algorithm for preventing starvation of computer resources to those work items which have been delayed, whereby the processing of all the work items in a program will be completed.

14. The program product of claim 13 wherein the priority algorithm increases respective valuations of delayed work items so as to complete processing of each of the work items prior to or at a cut-off processing date of the work item.

15. A networked environment, comprising:

a grid of computing resources;

a request manager of the grid to receive requests of one or more customers for utilization of computing resources of the grid;

one or more computer systems of a customer coupled to the request manager; the one computer system comprising one or more processors;

a memory coupled to at least the one processor of the one computer system; and,

a scheduling manager residing in the memory and executable by the at least one processor, the scheduling manager dynamically managing access of work items of a program to additional computer resources other than the at least one processor based on determining if valuations of each work item to be processed exceed estimated processor costs for respective ones of the work items.

16. The environment of claim 15 wherein the scheduling manager applies a valuation heuristic to each work item.

17. The environment of claim 15 wherein the scheduling manager delays each work item if an estimated processing cost of each work item exceeds the valuation of the respective work item.

18. The environment of claim 17 wherein the scheduling manager applies a priority algorithm for preventing starvation of computer resources to those work items which have been delayed, whereby the processing of all the work items in a program is completed.

19. The environment of claim 18 wherein the scheduling manager increases respective valuations of delayed work items so as to complete processing of each of the work items prior to or at a cut-off processing date of the work item.

20. A computer-implemented method for use in a networked environment including a grid of computing resources, and a request manager of the grid to receive requests of one or more customers for utilization of computing resources of the grid; wherein one or more computer systems of a customer is coupled to the request manager and include one or more processors; a memory coupled to at least the one processor; and, a scheduling manager residing in the memory and executable by the at least the one processor, comprising the steps of: dynamically managing the access of additional computer resources by respective ones of the work items if the valuation of each of the work items exceeds the cost of processing corresponding ones of the work items.

21. The method of claim 20 further comprising applying a valuation heuristic to each work item.

22. The method of claim 20 further comprising applying a priority algorithm for preventing starvation of computer resources to those work items which have been delayed, whereby the processing of all the work items in a program is completed.

23. A method of providing fee-based processing for programs in a processor system, whereby fees are based on projected utilization of computer resources to

be used for completing processing a program, the processor system including at least one processor; a memory coupled to the at least one processor, and a scheduling manager residing in the memory, the method comprising the steps of having the scheduling manager being executable for dynamically managing the access of additional computer resources to be applied to a program based on valuations of work items of a program that is to be processed; and, a mechanism for predicting costs for the dynamically determined computer resources to be used.

24. The method of claim 23 further comprising applying a valuation heuristic to each work item for establishing the valuation of each work item.

25. The method of claim 24 further comprising applying a priority algorithm for preventing starvation of computer resources to those work items which have been delayed, whereby the processing of all the work items in a program is completed.

26. The method of claim 25 wherein the dynamic determination is based on different attributes of the one or more work items forming at least part of a program.

27. A computer program product for use in a computer-implemented process for providing fee-based dynamic allocations of computer resources for executing a program, the computer program product comprising: a medium readable by a computer and having computer program adapted for: providing a scheduling manager being executable for dynamically managing the access of additional computer resources to be applied to work items of a program based on valuations of the work items; and, a mechanism for predicting costs for the dynamically determined computer resources to be used.